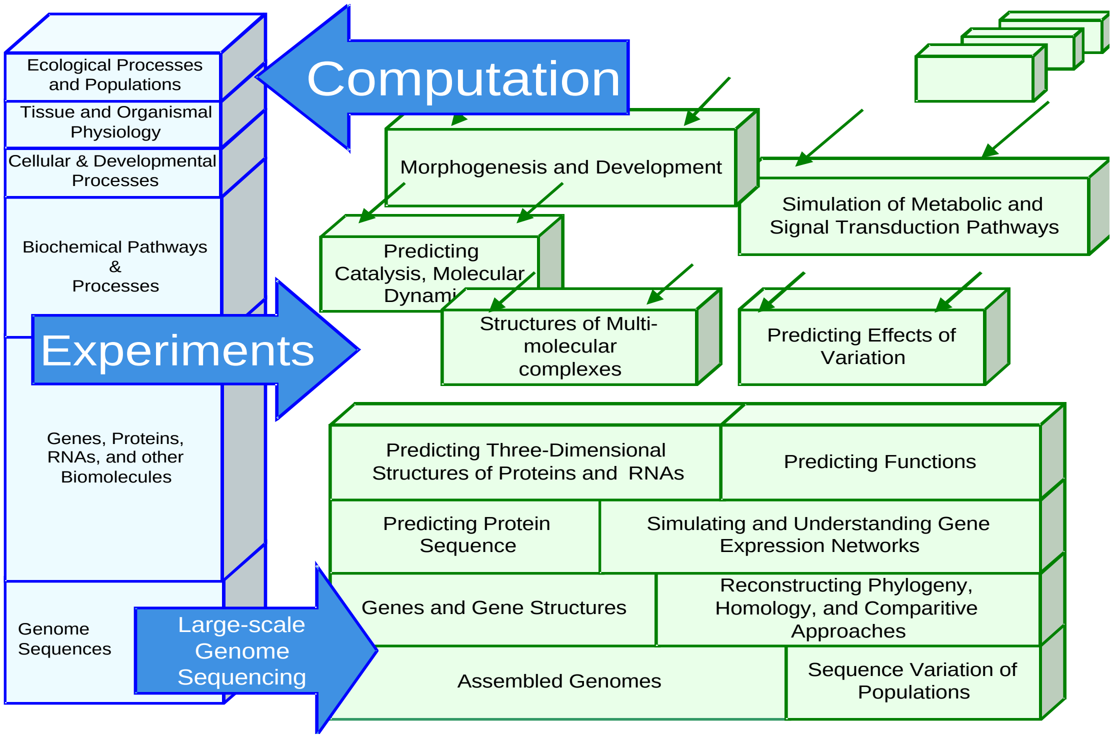
# Chapter 2
# High-Throughput Genome Assembly, Modeling, and Annotation

The sequencing of microbial genomes containing several thousand genes and the eventual completion of human and other model organism genomes is the underlying driving force for understanding biological systems at a whole new level of complexity. There is for the first time the potential to understand living organisms as whole, complex dynamic systems and to use large-scale computation to simulate their behavior. Modeling *all* levels of biological complexity is well beyond even the next generation of Teraflop computers, but each increment in the computing infrastructure makes it possible to move up the biological complexity ladder and solve previously unsolvable classes of problems.

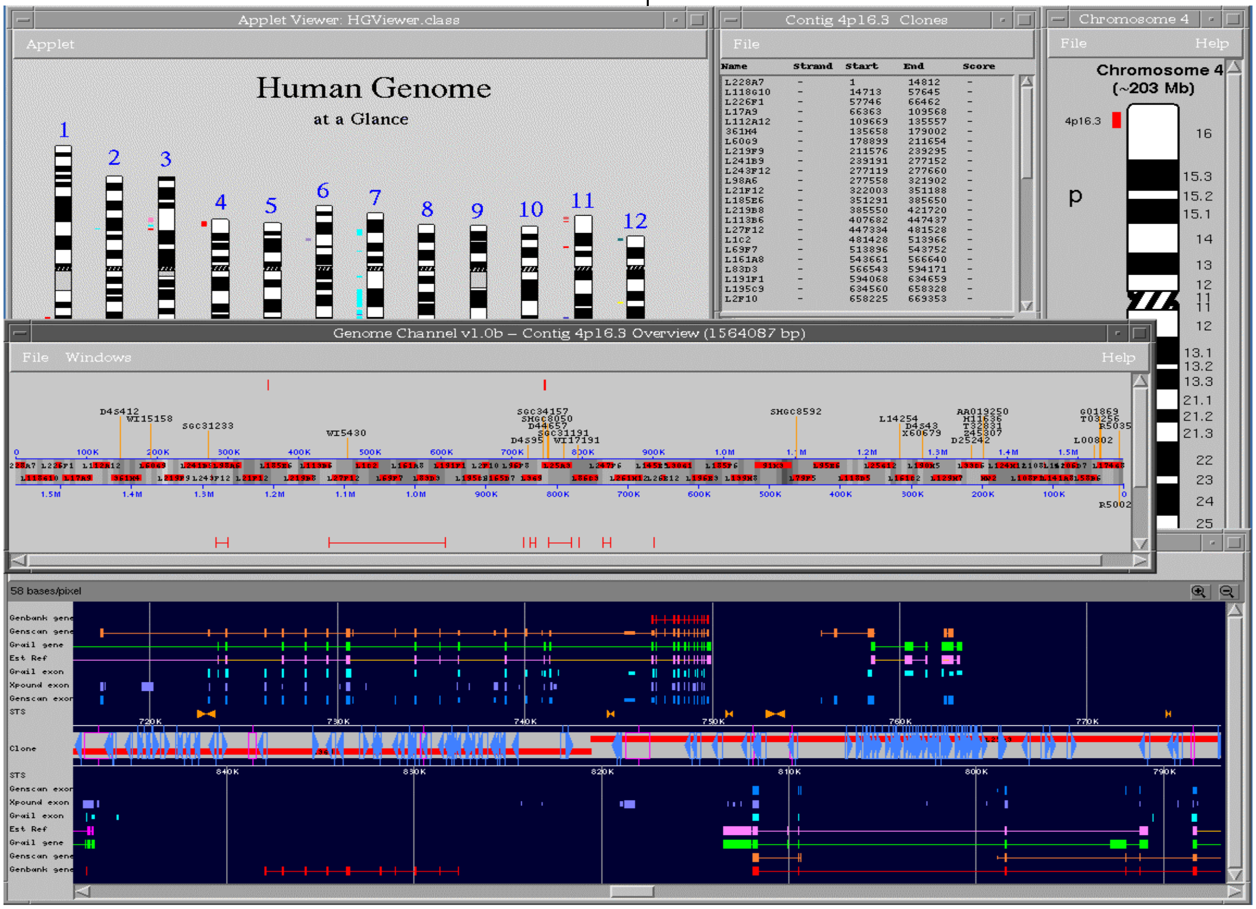

Ecological Processes and Populations
Tissue and Organismal Physiology
Cellular & Developmental Processes
Biochemical Pathways & Processes
Genes, Proteins, RNAs, and other Biomolecules
Genome Sequences
Computation
Experiments
Large-scale Genome Sequencing
Morphogenesis and Development
Simulation of Metabolic and Signal Transduction Pathways
Predicting Catalysis, Molecular Dynami
Structures of Multi-molecular complexes
Predicting Effects of Variation
Predicting Three-Dimensional Structures of Proteins and RNAs
Predicting Functions
Predicting Protein Sequence
Simulating and Understanding Gene Expression Networks
Genes and Gene Structures
Reconstructing Phylogeny, Homology, and Comparitive Approaches
Assembled Genomes
Sequence Variation of Populations

***The experimental (left) and computational (right) hierarchies will increasingly become codependent as the research community models greater biological complexity.***

The first step in the biological hierarchy is a comprehensive genome-based analysis of the rapidly emerging genomic data. With changes in sequencing technology and methods, the rate of acquisition of human and other genome data over the next few years will be ~100 times higher than originally anticipated. Assembling and interpreting these data will require new and emerging levels of coordination and collaboration in the genome research community to develop the necessary computing algorithms, data management and visualization systems.

Annotation− the elucidation and description of biologically relevant features in the sequence− is essential in order for genome data to be useful. The quality with which annotation is done will have direct impact on the value of the sequence. At a minimum, the data must be annotated to indicate the existence of gene coding regions and control regions. Further annotation activities that add value to a genome include finding simple and complex repeats, characterizing the organization of promoters and gene families, the distribution of G+C content, and tying together evidence for functional motifs and homologs.

As complete genomes are sequenced, the length of DNA comparison strings will change from single genes to entire genomes, with a concomitant expansion in the time to compute. In order to look at long-range patterns of expression, syntenic regions on the order of 10's of megabases become reasonable lengths for consideration. While the cycles needed to run many of these classes of analysis codes on rapidly increasing data sets is in itself a significant problem, all of these must inevitably be run on ALL data accumulated and in a recurring manner. Significant computational work is required to permit the analysis and visualization of long genomic regions needed for comparative genomic studies.



Contig 4p16.3 Clones

File

| Name | Strand | Start | End | Score |
|---|---|---|---|---|
| L228A7 | - | 1 | 14812 | - |
| L118G10 | - | 14713 | 57645 | - |
| L226F1 | - | 57746 | 66462 | - |
| L17A9 | - | 66363 | 109568 | - |
| L112A12 | - | 109669 | 135557 | - |
| 361H4 | - | 135658 | 179002 | - |
| L60G9 | - | 178899 | 211654 | - |
| L219F9 | - | 211576 | 239295 | - |
| L241B9 | - | 239191 | 277152 | - |
| L243F12 | - | 277119 | 277660 | - |
| L98A6 | - | 277558 | 321902 | - |
| L21F12 | - | 322003 | 351188 | - |
| L185E6 | - | 351291 | 385650 | - |
| L219B8 | - | 385550 | 421720 | - |
| L113B6 | - | 407682 | 447437 | - |
| L27F12 | - | 447334 | 481528 | - |
| L1C2 | - | 481428 | 513966 | - |
| L69F7 | - | 513896 | 543752 | - |
| L161A8 | - | 543661 | 566640 | - |
| L83D3 | - | 566543 | 594171 | - |
| L191F1 | - | 594068 | 634659 | - |
| L195C9 | - | 634560 | 658328 | - |
| L2F10 | - | 658225 | 669353 | - |

***The Genome Channel Browser.*** One mechanism for researchers to access and visualize the results of each day’s analysis of human genome data. The effort makes daily use of multiple current high-performance computing systems to keep up with current data flow and analysis and modeling requirements.

## Methods of Genome Sequence Analysis and Modeling

A new approach to sequencing genomes, whole genome shotgun sequencing, becomes possible if small fragments of sequence generated at random can be compared to each other fast enough to effectively determine sequence overlap and from this, assemble the pieces into longer contiguous regions, or contigs.

The first step in the sequence reconstruction involves finding overlaps between each fragment and the existing contigs. The presence of sequencing errors, natural sequence variations between sources as well as high repetitive DNA sequence elements make this an extremely complex challenge. If the computation can be performed and other significant technical problems can be overcome, this strategy could speed up sequencing of the human genome from about 7 to 3 years.

As an example, the proposed whole genome shotgun strategy at the Institute for Genome Research (TIGR) produced 30 million bases of DNA per day in January of 1998 which increased to 100 million base pairs per day by mid-year. To continue to grow the contiguous assemblies of sequence that emerge from the random strategy requires that each of the 200,000 fragments generated each day be compared in a very detailed way to all previous contigs. Particularly in the early phases, overlaps were rare so initially 200,000 contigs per day accumulated in the database. After enough contigs accumulated to realize substantial overlap, the sequence assembly required $4x10^{12}$ sequence comparisons per day, each of which requires a significant fraction of a second on a standard processor. While comparing new reads to existing data is enough of a problem, periodically the entire data structure must be rebuilt, requiring an even larger number of FLOPs for a calculation that breaks existing contigs down to their underlying fragments and reassembles them

Some of the most compute intensive processes involve the use of sequence database information to calculate models of genes contained in the sequence. In the last year or so, significant growth of expressed sequence tag (EST) collections and new types of hybrid gene modeling methods that integrate EST data with pattern recognition approaches, such as GRAIL-EXP, have become available. These can be used to produce modeling of the structure of genes in genome sequence data in most cases, but require significant computing power.

Nucleic acid based reference sequences currently number about 1 million and each putative exon (gene coding segment) must be aligned to this database of reference DNA pieces. GRAIL-EXP computes multiple-gene structures on an input DNA sequence using GRAIL predicted exons, which are most consistent with the known ESTs/cDNA/ protein sequences.

It achieves this goal by modeling the multiple-gene structure prediction problem as a combinatorial optimization problem and solving it by a dynamic programming method. The algorithm runs in $O(nM+n^2K^2)$ time, where n is the number of predicted exons in the given DNA sequence, M is the average time to find all the ESTt/cDNA/proteins that match a predicted exon from the specified databases, and K is the maximum number of EST/cDNAs/proteins an exon may match.

Typically, it takes about a minute to find all the matched ESTs in the current dbEST database (about 1.2 million entries) for a predicted exon. Hence it may take up to a few days to find all the matched ESTs for all the predicted exons on a DNA of 10 million bases long. If the data rate grows to on the order of 100 million bases per day, the calculation would require about a month of time for each day's data using a single 500 megahertz processor. This is by not a one time operation− the data needs to be reanalyzed frequently because the underlying databases of ESTs and cDNAs are growing rapidly. At 100 million bases per day, in ten days there is a billion base pairs to analyze, and in 100 days 10 billion. To reanalyze this data requires about three days on a current 1,024 node supercomputer. Bringing known protein sequences in the protein sequence database into the process makes the analysis much more useful, but due to the alignment takes an order of magnitude more time than the ESTs. Nonetheless, this should be done routinely, requiring about a month of time on a current 1024 node supercomputers.

The problem of parsing the predicted exons into genes that are most consistent with known ESTs/cDNAs/proteins, also takes significant amount of time, which will increase as the number of matched ESTs/cDNAs/proteins gets larger. To process 100 megabases of sequence will require over $10^{13}$Ops, assuming each predicted exon has ~1000 matched ests/ cDNA/ proteins in the database on average. When the average number of matched ESTs/cDNA/ proteins goes up to 5000, the number of needed ops will scale to over ~$10^{15}$Ops. It would also be desirable to calculate multiple gene models for each gene, to make each model consistent with possible splice variants suggested by the underlying EST evidence. This increases the complexity of this component of the calculation to $10^{17}$Ops for a relatively frequently needed operation.

## Methods for Large Scale Comparison of Genome Sequences

Once the basic structure of genes has been modeled, comparison of new sequences against each other or existing database is one of the most essential and revealing processes in computational genomics. Such operations relate new sequences to archival sequences that may have meaningful information about patterns in the sequence and its function. Such comparisons are the starting point for the computation of phylogenetic (evolutionary) trees of organisms or genes, pathogenicity studies for public health, polymorphism studies (e.g., of genetic defects), identification of protein motifs, model identification for gene recognition, model identification for organism classification, functional analysis of genomic/ protein sequences, and exon identification.

The analyses and inferencing often depend on the quality of the computed multiple sequence alignments (MSA's) used as input. MSA's of biological sequences, e.g., DNA, RNA, or protein sequences, entail the arrangement of many (in some cases thousands) of sequences, so that corresponding positions are aligned in vertical columns, with padding characters (nulls) added to compensate for length variations in some sequences.

The most accurate and sensitive alignments must consider gaps in the alignment (insertions and deletions) and are thus rather computationally intensive. The standard algorithm for this is Smith-Waterman, which uses dynamic programming to produce a local optimal alignment between two sequences of length M and N, and scales as O(MxN). The simple extension of these algorithms to multiple sequence alignments of K sequences requires time $O(N^K)$. For sequence lengths in the thousands of nucleotides, this is barely feasible for 3 sequences, certainly not for thousands of sequences. Hence, common practice is to use "progressive alignments" which is an inefficient algorithm that adds one sequence at a time to the MSA. This is computationally tractable, but not optimal. It is especially problematic when the sequences are not closely related, e.g., in computing the Tree of Life.

Recently rediscovered Hidden Markov Models (HMM) and Stochastic Context Free Grammars (SCFG) offer the prospect of better MSAs, by also modeling higher order structures. The simplest are HMMs, which are stochastic regular grammars. SCFGs are more complex, but permit one to model nested structures, such as the stem and loop structures common in RNA. More elaborate types of grammars permit the modeling of more complex secondary and tertiary structures. To use these models, one must first estimate the many parameters of the model. The resulting model can then be used to "parse" the sequences, and the resulting parses transformed into multiple sequence alignments. Iterative estimation of the Hidden Markov Models entails iterative solution of computations akin to the pairwise dynamic program sequence comparison computations. At each iteration we must perform M such $O(N^2)$ computations, one for each of the M sequences being aligned, and sum the results.

These independent computations with each sequence offers a clear target for parallel computation, followed by a logarithmic summation computation. This is particularly true for large sequence collections such as the ribosomal RNA alignments. Some researchers have constructed fine-grained parallel systolic algorithms for the dynamic programming computations, on specialized hardware implementations or SIMD machines. However, on MIMD machines (with greater costs for interprocessor communication and synchronization) coarser partitioning of the dynamic programming computations appears preferable. Furthermore, these iterative computations often find local optima, requiring multiple computations with different starting states to find (putative) global optima.

One difficulty in model estimation for methods like HMM arises from the possibility of over-fitting the very large number of parameters in these models (several per sequence position). Bayesian methods have been adopted to smooth these parameter estimates. Bayesian methods have traditionally been difficult to compute. Several researchers have resorted to Gibbs sampling methods to estimate the posterior probability distribution. These methods entail the construction and simulation of a Markov chain whose equilibrium probability distribution is equal to the target posterior distribution. The Gibbs sampling computations should be amenable to parallelization, assuming that independent parallel random number generators (PRNGs) are available. This is a subject of research activity in the Monte Carlo computation community, and are available from several research groups.

In the area of phylogenomics, insight from the evolutionary relationships of the unknown protein to others known is used to infer something about its potential function(s). In this approach, first, homologs of the unknown are identified and phylogenetic tree is constructed. Known functions of members of the group are overlaid onto the evolutionary tree and the function of the unknown is predicted by its position in the tree relative to its homologs whose functions have been characterized.

The first step in building a phylogenetic tree is to do a multiple sequence alignment on the homologous group of proteins. Once the alignment has been completed tree construction itself presents significant computational challenges. The evaluation and alignment of multiple trees, which is important for attempting to reconstruct the relationship among organisms based on several trees reflecting the relationships of groups of proteins, has been shown to be MAX SNP-hard.

For the "Tree of Life" computations that employ thousands of ribosomal RNA sequences, heuristic methods are a necessity. Typical computations with serial code run a few hundred hours on a workstation to accurately compute a single backbone tree of only about 100 nodes. Computation of the backbone tree involves both discrete optimization over the space of possible tree topologies and parametric optimization over the space of possible edge lengths (duration between evolutionary events). Likelihood computations are used to rank the trees. The likelihood computations are quite expensive, involving the computation of a state-vector for each node in the tree.

Divide and conquer strategies to partition the computation of the entire tree are also of interest− as the subtrees correspond to groups of related organisms. Each such taxa is typically of particular interest to a group of researchers. Bootstrap methods (resampling the input data and recomputation) can be to evaluate the reliability of the tree. Bootstrap computations exhibit obvious parallelism, but have previously been computational intractable for problems on this size. About a thousand new rRNA sequences are added to the "Tree of Life" every year.

A number of methods exist for the computation of phylogenetic trees. The area is one of considerable ongoing scientific controversy. These algorithms differ in the type of data used (distance data between molecules vs. aligned sequences), the existence of global optimal tree criteria, the existence of an explicit statistical model for the evolutionary history, etc. Methods vary in computation time, their consistency (convergence to correct tree with infinite data), efficiency (rate of convergence to correct tree with finite data), bias, robustness to departures from assumed statistical models of evolution.

Robustness is of particular concern because most analyses assume (contrary to fact) that individual sequence positions evolve independently. Efficiency of the estimation technique is particularly important when dealing with phylogenies of individual genes for which only relatively short sequences are available. Some studies have suggested that available data on individual genes may not be sufficient for reliable estimation of gene phylogenies.

Many researchers believe that maximum likelihood estimation (MLE) (or perhaps related Bayesian approaches), offer the best prospect for consistent, efficient estimation of phylogenetic trees. MLE approaches also facilitate the integration of multiple types of data (e.g., different sequences, restriction fragment length polymorphism data, etc.). However, MLE (and Bayesian) have formidable floating point computation and intermediate storage requirements (e.g., each internal node in the tree requires storage equal to the sequence length times the alphabet size). It is also worth noting that the large MLE phylogeny problems have serious problems with floating point representation of portions of the likelihood computations.

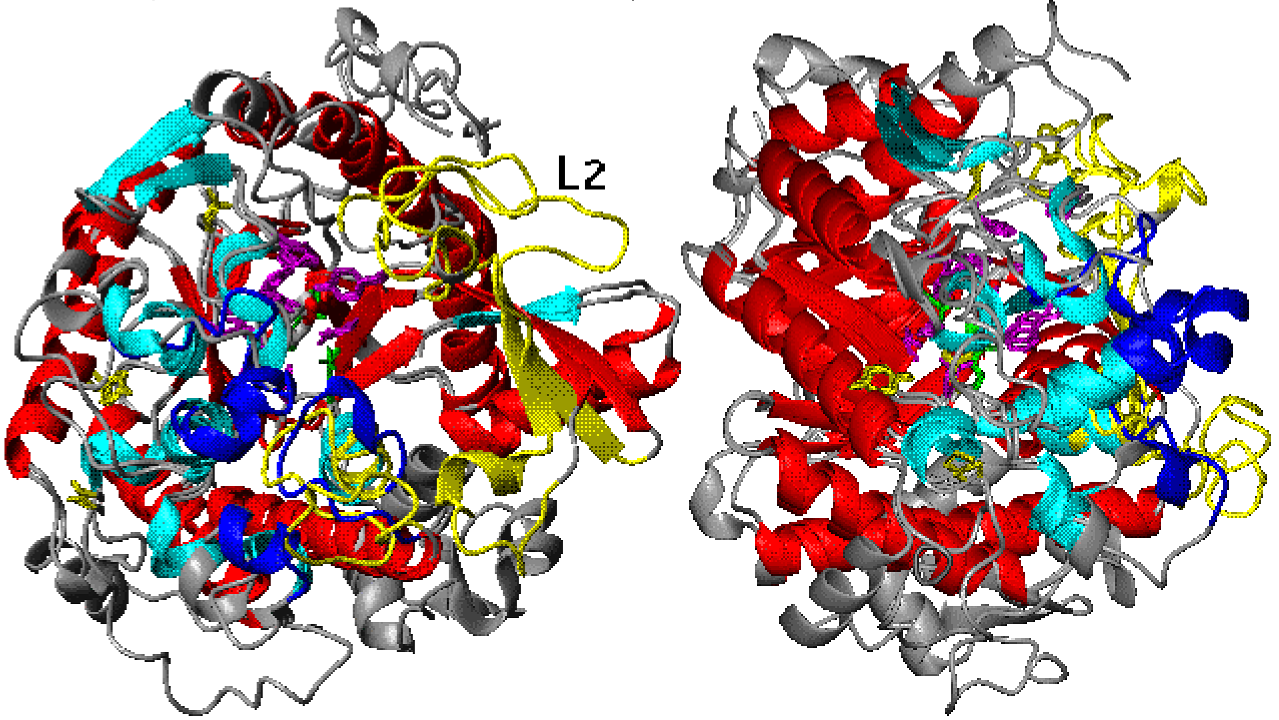
Clearly, some portions of the computations, e.g., bootstrapping, clearly lend themselves to simple cluster-based parallel processing. Finer grained parallelism, e.g., by decomposing the computation of individual trees, e.g., on MPPs, has yet to be explored. There also appear to be opportunities for parallelism in the computationally intensive construction of multiple sequence alignments, e.g., via HMMs or SCFGs.

# Sequence Comparisons Against Model Protein Families for Understanding Human Pathology

Searching a protein sequence database for homologues is a powerful tool for discovering the structure and function of a sequence. Amongst the algorithms and tools availabe for this task, Hidden Markov model (HMM)-based search methods improve both the sensitivity and selectivity of database searches by employing position-dependent scores to characterise and build a model for an entire family of sequences.

HMMs have been used to analyse proteins using two complementary strategies. In the first, a sequence is used to a search a collection of protein families, such as Pfam, to find which of the families it matches. In the second approach an HMM for a family is used to search a primary sequence database to identify additional members of the family. The latter approach has yielded insights into protein involved in both normal and abnormal human pathology such as Fanconi Anaemia A, Gaucher disease, Krabbe disease, polymyositis scleroderma and disaccharide intolerance II.

HMM-based analysis of the Werner Syndrome protein sequence (WRN) suggested it possessed exonuclease activity, and subsequent experiments confirmed the prediction. Like WRN, mutation of the protein encoded by the Klotho gene lead to a syndrome with features resembling ageing. However, Klotho is predicted to be a member of the family 1 glycosidase (see figure). Eventually, large-scale sequence comparisons against HMM models for protein families will require enormous computational resources to find these sequence-function correlations over genome-scale size databases.



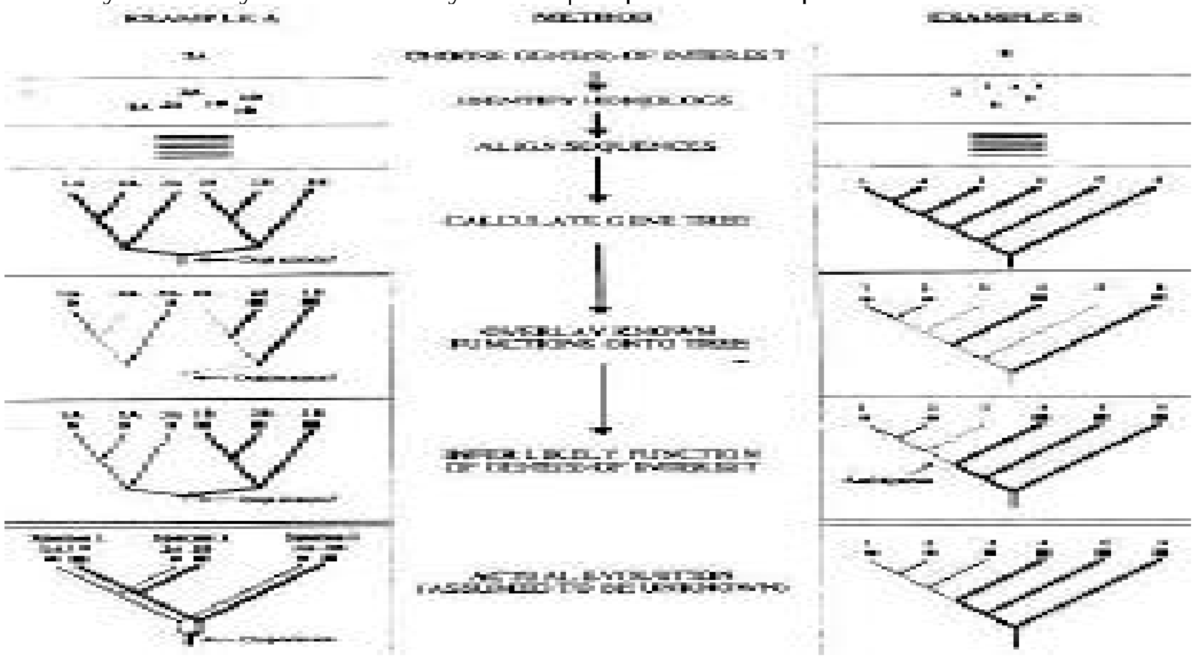
***The similarities and differences between two plant and archael members of a family of glycosidases that includes a protein implicated in ageing.*** Ribbons correspond to the beta-strands and alpha-helices of the underlying TIM barrel (red) and family 1 glycosidase domain (cyan). Amino acid side chains drawn in magenta, yellow and green are important for structure and/or function. The loop in yellow denotes a region proposed to be important for substrate recognition. The 2-deoxy-2-fluorglucosyl substrate bound at the active site of one of the enzymes is shown with carbon atoms in grey, oxygen in red and fluorine in green.

# Large-scale Calculations of Phylogenetic Trees

The calculation of phylogenetic trees is a central approach to understanding evolutionary history, a central problem in biology. Phylogenetic computations are concerned with the estimation of evolutionary trees and their reliability, and may be done from a variety of types of data: DNA sequences, ribosomal RNA sequences, protein sequences, or protein structures.

Currently construction of phylogenetic trees commences with a multiple sequence alignment (MSA), which is then used as input to the phylogenetic tree computation. Classical dynamic programming algorithms with progressive alignments can be used to do the MSAs. Hidden Markov models can improve the quality of the sequence alignments, and extensions to Stochastic Context Free Grammars have the ability to identify RNA secondary structures such as stem and loop construction.

Estimation of the phylogenetic tree itself involves searching the discrete topological space of all trees of a specified size (exponential in the number of leaves) and estimating the lengths of each edge in the phylogenetic tree (i.e. how much mutation occurred). Recent large phylogenetic computations have used $10^4$ to $10^5$ processor hours to explore truncated tree models. Typically, the reliability of the trees is estimated by performing a bootstrap computation over the individual sequence positions/ columns). For large computations, such as the tree of life, bootstrapping has often not been done, due to lack of computing resources. Speed and availability of computer time are presently major constraints of the ability to do these computations, as the availability of genome sequence data explodes.


EXAMPLE A
METHOD
EXAMPLE B
CHOOSE GENE(S) OF INTEREST
IDENTIFY HOMOLOGS
ALIGN SEQUENCES
CALCULATE GENE TREE
OVERLAY KNOWN FUNCTIONS ONTO TREE
INFER LIKELY FUNCTION OF GENE(S) OF INTEREST
ACTUAL EVOLUTION (ASSUMED TO BE UNKNOWN)

***Two hypothetical scenarios and the path of trying to infer the function of two uncharacterized genes in each case is traced.*** (A) A gene family has undergone a gene duplication that was accompanied by functional divergence. (B) Gene function has changed in one lineage. The genes are referred to by numbers representing the species from which these genes come, and letters representing different genes within a species. The thin branches in the evolutionary trees correspond to the gene phylogeny and the thick gray branches in A (bottom) correspond to the phylogeny of the species in which the duplicate genes evolve in parallel (as paralogs). Different symbols represent different gene functions; gray (with hatching) represents either unknown or unpredictable functions.

# The Need for High-End Computing for Genome Modeling and Annotation

The massive scale of the data flow and challenge of timely analysis is promoting significant changes in the organization of genome analysis components of the research community. A small number of specialized centers, such as the Genome Annotation Consortium, are emerging to construct the codes and systems needed to analyze the data on the scale needed for the next phase of the biological research. These groups are using current supercomputing capabilities on a continuing daily basis to keep up with current analysis and modeling needs and are rapidly recognizing the need for new computing infrastructure in the imminent future. Typical current applications, their current computational cost and community requirements are shown in Table I. These efforts are more and more serving as a focus for the broader research community by providing interface systems to access, visualize and validate the results obtained from genome scale computational analysis and modeling.

**Table I. Current and Expected Sustained Capability Requirements for Major Community Genomics Codes**

| Problem Class | Sustained Capability 1999 | Sustained Capability 2000 |
|---|---|---|
| Sequence assembly | >$10^{12}$ flops | $10^{14}$ flops |
| Binary sequence comparison | $10^{12}$ flops | >$10^{14}$ flops |
| Multiple sequence comparison | $10^{12}$ flops | >$10^{14}$ flops |
| Gene modeling | >$10^{15}$ flops | $10^{17}$ flops |
| Phylogeny trees | $10^{11}$ flops | $10^{13}$ flops |
| Protein family classification | >$10^{10}$ flops | $10^{12}$ flops |

**Table I** illustrates many high-priority computational challenges associated with the analysis, modeling and annotation of genome data. The first is the basic assembly and interpretation of the sequence data itself (analysis and annotation) as it is produced at increasing rates over the next five years. There will be a never-ending race to keep up with this flow, estimated at about 200 million base pairs per day by some time in 1999, and to execute the required computational codes to locate and understand the meaning of genes, motifs, proteins, and genomes as a whole. Cataloging the flood of genes and proteins and understanding their relationship to one another, their variation between individuals and organisms, and evolution represents a number of very complex computational tasks. Many calculations such as multiple sequence alignments, phylogenetic tree generation, and pedigree analysis are NP-hard problems and cannot be currently done at the scale needed to understand the body of data being produced, unless a variety of shortcuts is introduced. Related to this is the need to compare whole genomes to each other on many levels both in terms of nucleic acid and proteins and on different spatial scales. Large-scale genome comparisons will also permit biological inference of structure and function.